Computational Quantum Physics

Machine Learning Project

July 2025

Student: Christos Pilitsidis

Professor: Theodoros Diakonidis

Introduction/Methodology

In this project we are given a dataset of 2000 rows and 29 columns, that was acquired studying the Higgs field by supersymmetry theory. The first of the 29 columns is the label set and the rest are the feature sets(low level 1-21 and high level 22-28).

For the first part of the project i used two classifiers Random Forest and Decision Tree-two of the most popular Machine Learning algorithms for classification.The Decision Tree classifier has 'tree-like structure' of decisions where each node is a test on the features. On the other hand, the Random Forest algorithm is an ensemble of decision trees on random subsets of the data.

At first, an error message appeared, because the dataset contains a value that the python interpreter could not recognize as a number. The pandas library when 'reads' the csv file Higgs8K of the dataset 'sees' everything as string and then converts it to a float number. One entry could not be converted. I used the code in ErrorDetection.ipynb to find the error. Then i excluded the row of the error(first row) in everything that followed.

For the Decision Tree Model to avoid overfitting, i changed the depth of the tree(levels of questioning features). In the notebooks i have with comments each step of what i did. In every algorithm i checked for overfitting at the end. For the 3rd part of the project, i did the same with the only exception i included different columns for features(high level:22-28 and low level:1-21).

For Random Forest i changed the number of estimators(number of trees) and i tried to find a number that gives the best accuracy. The model was easily overfitting, so i had to change the depth of each decision tree. Details and comments for each step are given in each notebook.

For the second part of the project,i used the tensorflow library to create an Artificial Neural Network model for classification. Because we have binary classification(signal 1 or background 0), i used sigmoid function in the output layer and binary cross-entropy as a loss

function. I used trial and error method to find the number of hidden layers and number of neurons for each of them that gives a good accuracy score. At the end, i checked for overfitting. For the third part, i did the same but included different columns for features(high level:22-28 and low level:1-21).

Results/Comments

We have the following table:

Accuracy scores			
	All features	High	Low
Random Forest	0.67	0.678	0.5415
Decision Tree	0.6685	0.6635	0.5615
ANN	0.6525	0.6793	0.5681

All models perform better on high-level features than low-level features-which confirms that feature engineering is effective. Highest accuracy is achieved with ANN(Artficial Neural Network) on the high level features and the lowest accuracy is achieved with Random Forest on the low level features. For Random Forest and ANN models high level features slightly outperform all the others, suggesting some features in the full set may add noise or are redundant. Also Random Forest outperforms slightly Decision Tree on all feature set and it is expected since it averages multiple trees.